Discovering Patterns in Flows: a Privacy Preserving Approach with the ACSM Prototype

Stéphanie Jacquemont, François Jacquenet, Marc Sebban

Discovering Patterns in Flows: a Privacy Preserving Approach with the ACSM Prototype*

Stéphanie Jacquemont[1], François Jacquenet[1], and Marc Sebban[1]

Université de Saint-Etienne, Laboratoire Hubert Curien,
18 rue Benoît Lauras, 42000 Saint-Etienne, France

Abstract. In this demonstration, we aim to present the ACSM prototype that deals with the discovery of frequent patterns in the context of flow management problems. One important issue while working on such problems is to ensure the preservation of private data collected from the users. The approach presented here is based on the representation of flows in the form of probabilistic automata. Resorting to efficient algebraic techniques, the ACSM prototype is able to discover from those automata sequential patterns under constraints. Contrary to standard sequential pattern techniques that may be applied in such contexts, our prototype makes no use of individuals data.

1 Introduction

Sequential pattern mining has been a very active domain of research since the mid 90's (see [1] for an overview of the domain) and it has provided various powerful tools used in many applications. Nevertheless, those researches have mainly focused on increasing the efficiency of algorithms in terms of speed and space consumption. Parallel to this line of research, there has been a great interest in privacy preserving data mining techniques [2], but not directly dedicated to the specific domain of sequence mining, except the work of Zhan et al. [3], and more recently that of Kapoor et al. [4] or Kim et al. [5]. However, in those approaches, it is supposed that we have a set of sequences on which we can apply some specific techniques in order to do some kind of anonymization, directly or indirectly on the data to be processed.

We can note that there are many situations where the database of sequences results from the study of flows. For example, we may study the flow of cars in a town and get some set of paths used by car drivers on which the sequence mining task could be achieved. We may study the flow of visits on a particular web site and get some history files of sequences of pages visited by some users. We may study the flow of IP packets in some networks and get some set of routes used by IP packets sent and received by users. We may also study the flow of customers

* This work has been supported in part by the french national research agency under the Bingo2 project

in a store and get some video recordings of customer behaviors in the store, etc. In such situations, anonymization techniques may be used on the data with more or less difficulties. However, this would result in a huge amount of pre-processing and moreover, and would require to collect non anonymous datasets that might be used by malicious agents before the end of the anonymization process. The approach we propose with the ACSM (Automata-based Constrained Sequence Mining) prototype is completely different. We assume that the underlying problem can be modeled in the form of a probabilistic automaton. This hypothesis is particularly verified in flow management problems such as those previously mentioned. In such a context, ACSM achieves a sequence mining task directly from the automaton using algebraic methods. Hence, it does not need to get the sequences of data making the flow but rather the structure of the flow itself (an automaton) and the values assigned to each path of the flow.

2 The ACSM prototype

Due to space limitation we cannot present the theoretical aspects behind ACSM. For that purpose, the reader may refer to [6]. From an operational point of view, ACSM takes as input an XML file describing a flow graph. This file is loaded by ACSM that generates the corresponding probabilistic automaton while satisfying some statistical constraints.

ACSM can then be used to extract frequent sequential patterns given a support threshold. Our software has been implemented to satisfy statistical constraints in order to reduce the risk to extract false frequent patterns or to overlook true frequent ones. In this probabilistic framework, the user can then parameterize ACSM according to a priori fixed type I and type II errors. We dealt with the control of the false positive and false negative rates in [7] by providing a lower bound of the sample size on which the sequence mining task has to be performed. This means that we can formally decide when we can stop observing the flow, pick up the XML file and begin to mine the data.

The probabilities on the edges and vertices of the automaton are used to calculate the frequency of each discovered sequence and ACSM returns the paths (sequences), made up of not necessarily consecutive edges, whose frequency is greater than the fixed support threshold. Efficient algebraic calculus are used to optimize the extraction process, based on the LU factorization.

3 Overview of the demonstration

To put forward the interest of ACSM, we show an application (called TrafficMiner) in the domain of car flow management that has been designed on top of it. Using ACSM to discover car flow patterns in a town, that is sequences of non necessarily consecutive streets frequently used by car drivers of that town, is a way to ensure privacy preservation of drivers' behavior. Indeed, until now, if we wanted to discover frequent routes in a town we had to install cameras in each street and then record the routes traversed by each driver. This solution is

of course unrealistic because of its cost, but moreover it would be a great breach to drivers privacy and this would not be acceptable. The solution we propose consists in just counting, in each street of the part of the town we want to study, the number of cars using this street. The map of the town and the resulting counters provide a probabilistic automaton that can be processed by ACSM.

Figure 1 shows a screenshot of TrafficMiner discovering frequent routes in the city of Arlington (Virginia, USA).

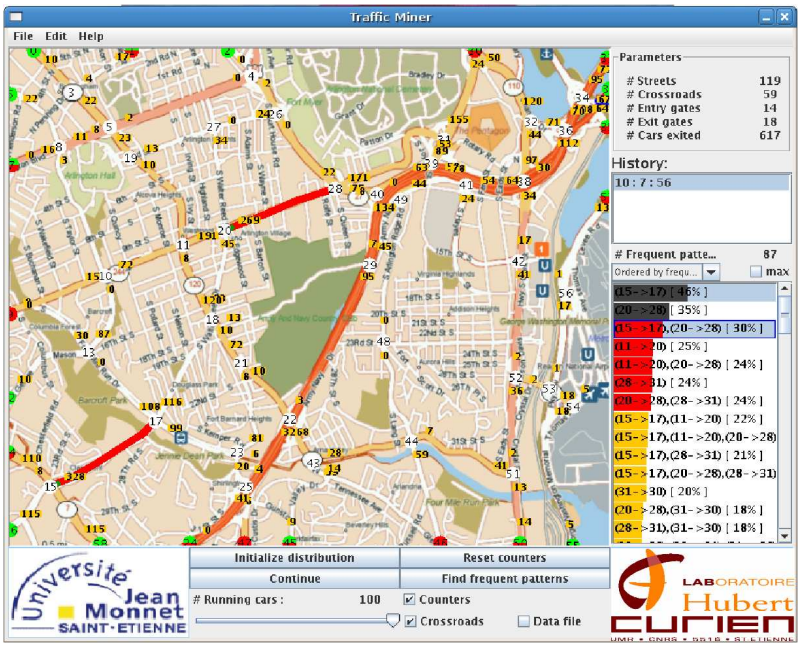

Fig. 1. TrafficMiner running on a map of Arlington.

The demonstration shows the way we can design the automaton associated with the map using a specific interface and the way we can simulate some traffic on this map. Then, we look at the XML file describing the flows of cars focusing on each of its markup. Then, we run ACSM to get the frequent routes used by the virtual car drivers.At the demo, it will also be possible for users to provide specific flows that may have some interests for them. The input data will have to be provided in an XML form explained on site. Then running ACSM will provide patterns for the users.

3.1 Contribution of ACSM for the ML and DM communities

What makes our piece of software unique and special? To our knowledge, ACSM is the first prototype able to deal with flow management problems and

to extract sequential patterns in datasets completely preserving the privacy of users involved in the flows.

What are the innovative aspects or in what way/area does it represent the state of the art? The innovative aspects of our system concerns privacy preservation issues. Contrary to standard sequential pattern mining algorithms, our prototype does not need any user dependent sequential data. No anonymization step is required because the sequential data are no more needed.

For whom is it most interesting/useful? Our prototype, and the ideas behind, may be very useful for the industrial practitioners. To give an example, we may consider webmasters who might want to mine visits of users without using classical history files that lead to many known problems due to proxies, caches, etc. Instead of a tedious pre-processing step on history files, they may easily proceed by putting counters on each page and link of their site and then use ACSM on the automaton built using the structure of the website and the probabilities assigned thanks to the counters.

4 Conclusion

We think this demonstration of the ACSM system proves the interest of processing automata instead of sequences in various situations. This is the case for example if privacy concerns are crucial issues of the applications. Interoperability with ACSM is very easy as it only requires an XML file as an input, which gives ACSM the ability to be easily encapsulated in some specific applications.

References

1. Dong, G., Pei, J.: Sequence Data Mining. Springer (2007)
2. Aggarwal, C., Yu, P.S., eds.: Privacy-Preserving Data Mining: Models and Algorithms. Springer (2008)
3. Zhan, J., Chang, L., Matwin, S.: Privacy-preserving collaborative sequential pattern mining. In: Proc. of the Workshop on Link Analysis, Counter-terrorism, and Privacy in conjunction with the SIAM ICDM. (2004) 61–72
4. Kapoor, V., Poncelet, P., Trousset, F., Teisseire, M.: Privacy preserving sequential pattern mining in distributed databases. In: Proc. CIKM. (2006) 758–767
5. Kim, S.W., Park, S., Won, J.I., Kim, S.W.: Privacy preserving data mining of sequential patterns for network traffic data. Information Sciences **178**(3) (2008) 694–713
6. Jacquemont, S., Jacquenet, F., Sebban, M.: Mining probabilistic automata: A statistical view of sequential pattern mining. Machine Learning Journal **75**(1) (April 2009) 91–127
7. Jacquemont, S., Jacquenet, F., Sebban, M.: A lower bound on the sample size needed to perform a significant frequent pattern mining task. Pattern Recognition Letters (2009) To appear.